



SMART CONTRACTS

BASIC CONCEPTS

- Smart Contract is an **agreement**
 - define the rules and penalties
 - automatically enforce those obligations
- Smart Contract is a **program**
 - the code runs at some point
 - automatically validates conditions and do the processing

EXAMPLE – DIGITAL NOTARY

- Idea

*create a digital notary that stores fingerprint of documents as
proofs of their existence*

- Solution

*use Ethereum platform to create a **Ɗapp***



ETHEREUM – UNDER THE HOOD

- Ethereum was designed as a smart contract **platform**
- EVM – **distributed** global computer where all smart contracts are executed
- Using **gas** to limit the resources used by each contract
- Each operation costs gas **paid** in Ether
- Ethereum accounts pays for transaction but sets **gas limit**

THE CONTRACT

- Created with a programming language
 - Solidity, LLL, Serpent
- Has state and functions
- Interaction:
 - function calls or events


```
1  pragma solidity ^0.4.4;
2
3  // Proof Of Existence smart contract
4  // create a digital notary that stores hashes of documents
5  // as proofs of their existence
6  contract ProofOfExistence {
7      // contract state as a mapping data structure
8      mapping (bytes32 => bool) private proofs;
9
10     // Function to store a proof of existence in the contract state
11     function storeProof(bytes32 proof) public {
12         proofs[proof] = true;
13     }
14
15     // Function to calculate and store the proof for a document
16     function notarize(string document) public {
17         // calculate proof
18         var proof = proofFor(document);
19         // store proof in a contract state
20         storeProof(proof);
21     }
22
23     // Utility function to get a document's hash (sha256 algorithm)
24     // Note: read-only function - not changing the state
25     function proofFor(string document) public pure returns (bytes32) {
26         return sha256(document);
27     }
28
29     // Function to check whether document was stored in a blockchain
30     // Returns true if proof is stored, false otherwise
31     function hasProof(bytes32 proof) private constant returns(bool) {
32         return proofs[proof];
33     }
34
35     // Function to check if a document has been notarized
36     // Returns boolean
37     function checkDocument(string document) public constant returns (bool) {
38         var proof = proofFor(document);
39
40         return hasProof(proof);
41     }
42 }
```

POE - HANDS ON EXPERIENCE

Digital Notary

A simple Dapp to store hashes of documents as proofs of their existence.

Document hash: 5f090668547ee6d5b36253563f8ec97d788b025b5d4a85eb02c2f32bf97e695f

 Clear content

LegalTech Polska 2018

CONNECTION



ACCOUNT

0x6273..ef57

BALANCE

99.80315382 ETH

LATEST BLOCK NUMBER

66

COLLATED AT

02/14/2018 14:50

Gas limit

421,000

 Deploy contract

 Notarize

 Check document

 Get signature

IS IT AWESOME?

- Autonomy – execution managed automatically
- Trust – documents are encrypted on a shared ledger
- Backup – documents are duplicated many times over
- Safety – encryption everywhere
- Speed – software code that automates tasks
- Saving – depending on use-case money may be saved
- Accuracy – avoiding (human) errors

... IT'S NOT PERFECT

- It is still the code
 - ... and still has bugs
- It is not trivial
 - experienced staff needed to make the job done right
- Evolving technology
 - work in progress in many knowledge domains
- Stack
 - supplemented by technologies running the blockchain



DZIĘKUJEMY ZA UWAGĘ

tomasz.fidecki@aply.eu
pawel.slawacki@aply.eu